

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1 1. (Currently amended) A method for creating a plurality of queues ~~using~~
2 within a shared data buffer, the method comprising:
3 providing a plurality of pointers to the data buffer, each pointer associated
4 with an area of the buffer; and
5 creating a given queue in the plurality of queues by associating a given
6 pointer from the plurality of pointers with the given queue;
7 wherein a given area of the data buffer can be assigned to the given queue
8 and then reassigned to a different queue in the plurality of queues at a later time.

1 2. (Original) A method according to claim 1, wherein providing a plurality
2 of pointers includes storing the plurality of pointers in a free pointer linked list.

1 3. (Original) A method according to claim 2, wherein associating the given
2 pointer includes removing the given pointer from the free pointer linked list.

1 4. (Original) A method according to claim 3, wherein associating the given
2 pointer further includes storing the pointer in a given queue linked list.

1 5. (Original) A method according to claim 4 further including:
2 removing the given pointer from the queue linked list and adding the given
3 pointer to the free pointer linked list to delete a member of the given queue.

1 6. (Original) A method according to claim 5, wherein the given queue is a
2 FIFO queue.

1 7. (Original) A method according to claim 5, wherein the given queue is a
2 LIFO queue.

1 8. (Original) A method according to claim 4 wherein the free pointer
2 linked list and the given queue linked list are stored in a given data structure.

1 9. (Currently amended) A computer program product for use on a
2 computer system for managing a plurality of queues, ~~employing within~~ a shared
3 data buffer, the computer program product comprising a computer usable medium
4 having computer readable program code thereon, the computer readable program
5 code including program code for:
6 providing a plurality of pointers to the data buffer, each pointer associated
7 with an area of the buffer; and
8 creating a given queue in the plurality of queues by associating a given
9 pointer from the plurality of pointers with the given queue;
10 wherein a given area of the data buffer can be assigned to the given queue
11 and then reassigned to a different queue in the plurality of queues at a later time.

1 10. (Original) A computer program product according to claim 9, wherein
2 providing a plurality of pointers includes storing the plurality of pointers in a free
3 pointer linked list.

1 11. (Original) A computer program product according to claim 10,
2 wherein associating the given pointer includes removing the given pointer from
3 the free pointer linked list.

1 12. (Original) A computer program product according to claim 11,
2 wherein associating the given pointer further includes storing the pointer in a
3 given queue linked list.

1 13. (Original) A computer program product according to claim 12 further
2 including:
3 removing the given pointer from the queue linked list and adding the given
4 pointer to the free pointer linked list to delete a member of the given queue.

1 14. (Original) A computer program product according to claim 13,
2 wherein the given queue is a FIFO queue.

1 15. (Original) A computer program product according to claim 13,
2 wherein the given queue is a LIFO queue.

1 16. (Original) A computer program product according to claim 12 wherein
2 the free pointer linked list and the given queue link list are stored in a given data
3 structure.

1 17. (Currently amended) A device for managing a plurality of queues in a
2 computer system, the device comprising:
3 a shared data buffer;
4 a pointer array pointing to a plurality of areas of the data buffer, wherein a
5 given area of the data buffer can be assigned to the given queue and then
6 reassigned to a different queue in the plurality of queues at a later time;
7 a free list data structure including an entry count, a head pointer to the data
8 buffer and a tail pointer to the data buffer;

9 a queue state including a plurality of virtual queue data structures, each
10 queue data structure including a queue entry count, a queue head pointer and a
11 queue tail pointer, the queue head pointer and the queue tail pointer pointing to
12 areas of the data buffer; and
13 logic for deleting an entry from the free list data structure and adding the
14 entry to a given virtual queue data structure.
15

1 18. (Original) A device according to claim 17, the device further
2 comprising:
3 logic for deleting an entry from a given virtual queue data structure and
4 adding the entry to the free list data structure.